Sistema de videovigilancia a través de una Raspberry Pi

Aitor Domene-Sánchez

Resumen—En este artículo se explica el desarrollo de un sistema de videovigilancia utilizando un ordenador de placa reducida, la Raspberry Pi. Este proyecto tiene el objetivo de producir y distribuir un sistema de videovigilancia con capacidades avanzadas y a la vez que tenga un bajo coste. Mediante un Smartphone con la aplicación Telegram instalada, cualquier persona con conexión a internet podrá monitorizar lo que esté ocurriendo en el lugar dónde esté instalado el sistema, permitiendo al usuario observar todo mediante imágenes y en streaming, y produciendo alertas si detecta movimiento. Este proyecto intenta acaparar el mercado para un sistema de videovigilancia barato y fácilmente configurable en el que se puede aplicar en una gran variedad de entornos con diferentes ventajas y objetivos. En este caso, está destinado a un entorno particular y personal ofreciéndole al usuario una herramienta más para la seguridad de su hogar.

Palabras clave—Raspberry Pi, Telegram, Bot, Streaming, Chatbots, Pi Camera, Buzzer, PIR, IR Remote, Servo Motor, Token, Wireframes, Clave Pública, Socket, OTP

Abstract—This article explains the development of a video surveillance system using a single-board computer, Raspberry Pi. This project aims to produce and distribute video surveillance system with advanced capabilities and at the same time has a low cost. With a Smartphone and with the Telgram application installed, anyone with an internet connection can monitor what is happening in the place where the system is installed, allowing the user to observe what it captures by streaming and producing alerts if it detects movement. This project tries to hoard the market for a cheap and easily configurable video surveillance system in which it can be applied in a wide variety of environments with different advantages and objectives. In this case, it is intended for a particular environment and personal offering the user an additional tool for the safety of your home.

Keywords— Raspberry Pi, Telegram, Bot, Streaming, Chatbots, Pi Camera, Buzzer, PIR, IR Remote, Servo Motor, Wireframes, Toke, Public Key, Socket, OTP

1. Introducción

Los actos delictivos en nuestro país aumentan, por lo que es necesario tomar medidas de precaución para nuestra seguridad. En los últimos años los robos en viviendas se han incrementado en un 73% [1]. Desde el inicio de la crisis en España en 2009, más de tres millones de viviendas han sido asaltadas en nuestro país.

La instalación de un sistema de videovigilancia permite al ciudadano sin duda una sensación de seguridad y tranquilidad mucho mayor. Uno de los principales atractivos de un sistema de videovigilancia es la posibilidad de estar conectados con nuestro hogar las 24 horas del día. De este modo, desde cualquier lugar, el ciudadano tiene la tranquilidad de saber todo lo que pasa en su hogar en cualquier momento, manteniendo una supervisión constante.

- *E-mail de contacte: aitor.domene@e-campus.uab.cat*
- *Menció realitzada: Enginyeria del Software.*
- *Treball tutoritzat per: Fernano Vilariño (CC. de la Computación)*
- *Curs 2016/17*

Sin embargo, tener esa seguridad para el hogar tiene un cierto coste. Las empresas de seguridad ofrecen sistemas de videovigilancia en régimen de alquiler. El precio de sus packs parte de los 200€ aproximadamente hasta los 1.000€, a lo que hay que sumar las cuotas mensuales de entre 25 y 40 euros [1].

Para poder dar una alternativa más económica al ciudadano nace la propuesta de este TFG *Sistema de videovigilancia a través de una Raspberry Pi*. La finalidad de este trabajo es desarrollar un sistema de videovigilancia con avanzadas opciones manteniendo un bajo coste respecto a las empresas de seguridad.
Esto será posible mediante el uso la *Raspberry Pi* [2], el micro ordenador más utilizado en todo el mundo. Las posibilidades de este dispositivo son inmensas, desde convertirlo en una retro-consola, un servidor local o, en este caso, un sistema de videovigilancia, entre otros.

Este micro ordenador permitirá al ciudadano tener un sistema de seguridad en su hogar en el que él mismo pueda gestionarlo desde su Smartphone con la aplicación *Telegram [3],* un servicio de mensajería. Este sistema se complementa de una segunda Raspberry Pi para crear un servidor de página web y permitir al usuario acceder a las imágenes y vídeos que el sistema de videovigilancia haya guardado.

Con el uso de diferentes componentes electrónicos que utilizará la Raspberry Pi, el sistema de videovigilancia se convierte en un sistema con opciones avanzadas:

- Detección de movimiento.
- Notificación de intruso desde el servicio de mensajería Telegram junto a una imagen y vídeo del intruso.
- Alarma sonora al detectar intruso.
- Solicitar captura de imagen desde Smartphone (vía Telegram).
- Solicitar vídeo streaming desde Smartphone (vía Telegram).
- Consultar historial de imágenes y videos del sistema desde la plataforma web.
- Realizar vídeo streaming desde la plataforma web.
- Posibilidad de rotar la cámara del sistema.
- Posibilidad de activar o desactivar el sistema desde el Smartphone (vía Telegram) o mando a distancia.

Con estas características y el bajo consumo de la Raspberry Pi, convierten al sistema en un producto competitivo en seguridad del hogar.

La estructura del resto del artículo se organiza de la siguiente manera: en la Sección 2 se hará un análisis del estado actual del arte; viendo los diferentes proyectos de sistema de videovigilancia basados en una Raspberry Pi. La Sección 3 hablará de los objetivos del proyecto. Las Secciones 4 a la 7 recorrerán partes vitales del proceso de desarrollo del software. En ellas, se hablará de la metodología seguida para la realización del trabajo y del por qué se ha escogido, la selección de requerimientos y la planificación organizada para el desarrollo del trabajo, las tecnologías utilizadas, y la arquitectura hardware y software del sistema. Las Secciones 8 a la 10 se hablarán del desarrollo del trabajo, con las diferentes pruebas de software realizadas a lo largo del proyecto y los resultados del trabajo. Finalmente, la Sección 11 servirá como conclusión del artículo. En ella, se resumen los principales puntos tratados y las futuras líneas de mejoras del proyecto.

2 Estado del arte

En la actualidad, existen múltiples proyectos de sistemas de videovigilancia realizados con la Raspberry Pi, cada uno de ellos con ventajas e inconvenientes. He realizado un estudio de ejemplos paradigmáticos para hacer un análisis de la competencia y ver las diferencias entre los distintos proyectos. Un conjunto de proyectos que ataca un sistema de videovigilancia low-cost, son los siguientes:

- Sistema de videovigilancia basado en Raspberry Pi que monitoriza lo que está sucediendo en el lugar donde esté instalado el sistema [4]. Este sistema permite alertas a través de correo electrónico si detecta movimiento. El gran inconveniente de este proyecto es la incomodidad para gestionar el sistema. El usuario debe de estar en todo momento delante de un PC para poder monitorizar el sistema.
- Un sistema de videovigilancia basado en una Raspberry Pi que permite visualizar y controlar el acceso a una vivienda [5]. Este sistema se controla mediante una página web.

Cabe destacar que estos dos proyectos mencionados, y muchos otros más, son sistemas que permiten conectarse a ellas y visualizar lo que sucede. Pero por una serie de inconvenientes no logran ser un sistema de videovigilancia potente. La importancia de ser notificado de inmediato adjuntando pruebas, gestionar el sistema desde un Smartphone y almacenar pruebas si el sistema de videovigilancia detectase algún caso inusual es motivo por el cual este trabajo se considera un sistema de videovigilancia con capacidades avanzadas.

3 Objetivos

Este trabajo trata de crear un un sistema de videovigilancia gestionado por la Raspberry Pi para facilitarle al ciudadano un sistema de seguridad completo y de bajo coste. Para lograr este propósito, se definieron los siguientes objetivos:

- Gestionar diferentes componentes electrónicos conectados a la Raspberry Pi para proporcionar un sistema con capacidades avanzadas.
- Implementar un sistema de interacción con el sistema para el usuario
- Integrar una comunicación entre el sistema y el servicio de mensajería Telegram.
- Integrar una plataforma web con un historial del sistema y un video streaming.

4 Metodologia

Para desarrollar este proyecto se ha hecho uso de la metodología *Feature Driven Development* [6]. La metodología Feature Driven Development, ha permitido que el proyecto se haya desarrollado con un software de calidad y con un monitoreo constante del proyecto, asegurando que cada una de las características que se han ido desarrollando hayan estado bien diseñadas y testeadas. Las etapas de esta metodología aplicadas al proyecto han sido: elaborar, planificar, diseñar y desarrollar funcionalidades.

Para redactar la documentación necesaria y tener el trabajo de desarrollo organizado y planificado he utilizado el concepto *sprint* o *iteración* [7] pertenecientes a las metodologías ágiles.

5 Requerimientos del sistema

La primera tarea realizada ha sido la creación de una lista de requerimientos a cumplir del proyecto.

Requerimiento	Prioridad
El sistema debe de proporcionar al usuario un menú de opciones intuitivo para interactuar con el sistema.	**Medium**
El sistema debe enviar una captura de imagen si el usuario lo solicita vía Telegram.	**High**
El sistema debe ejecutar un video streaming si el usuario lo solicita vía Telegram.	**High**
El sistema debe activarse/desactivarse si el usuario lo solicita vía Telegram.	**High**
El sistema debe activarse/desactivar mediante un control remoto.	**Medium**
El sistema debe de notificar al usuario vía Telegram si detecta una presencia.	**High**
El sistema debe de enviar una captura de imagen si detecta una presencia vía Telegram.	**High**
El sistema debe de enviar un video de 5 segundos si detecta una presencia vía Telegram.	**High**
El sistema debe ejecutar una alarma sonora si detecta una presencia.	**High**
El sistema debe rotar la cámara a petición del usuario.	**Medium**
El sistema debe enviar una captura de imagen al servidor web al detectar una presencia.	**High**
La plataforma web debe de tener un sistema de autentificación para el acceso.	**High**
El sistema debe enviar un video al servidor web al detectar una presencia.	**High**
La plataforma web debe proporcionar un historial de imágenes y videos.	**High**
La plataforma web debe proporcionar un video streaming.	**High**

Tabla 1. Requerimintos del sistema

6 PLANIFICACIÓN DEL TRABAJO

Desde las primeras fases del proyecto ya estaba claro qué puntos se querían tratar. Así pues, el proyecto se ha dividido en 2 bloques, el desarrollo de las funcionalidades del sistema y el desarrollo de la plataforma web; en los cuales se han realizado diversos *sprints* de una o dos semanas.

El primer bloque se ha llevado a cabo la implementación de las funcionalidades del sistema de videovigilancia. Se ha trabajado con la comunicación entre la aplicación Telegram y la Raspberry Pi, y los diferentes componentes electrónicos. Por otro lado, el segundo bloque se ha destinado a la creación de la plataforma web, tanto *back-end* como *front-end*, y la implementación de la comunicación entre el sistema de seguridad y la plataforma web.

A continuación, se exponen los sprints llevados a cabo en este trabajo:

Bloque 1:

- **Sprint 0**: Instalación y configuración de los diferentes recursos necesarios para la Raspberry Pi
- **Sprint 1**: Implementar comunicación entre Raspberry Pi y Telegram
- **Sprint 2**: Implementación Pi Camera
- **Sprint 3**: Implementación Sensor PIR
- **Sprint 4**: Implementación Buzzer
- **Sprint 5**: Análisis e implementación de nuevos componentes

Bloque 2:

- **Sprint 6**: Diseño de la plataforma web
- **Sprint 7**: Implementación plataforma web
- **Sprint 8**: Implementar comunicación sistema de videovigilancia y plataforma web

En el Apéndice 1 se puede observar el diagrama de Gantt del proyecto para visualizar en detalle la planificación.

7 COMPONENTES DEL SISTEMA

Una vez definidos los requerimientos del sistema y profundizados en ellos, se ha llevado a cabo la selección del Hardware y Software a utilizar para el desarrollo del proyecto y la generación y mantenimiento de la documentación.

7.1 Hardware

7.1.1. Componentes electrónicos

Para desarrollar un sistema de videovigilancia con capacidades avanzadas con la Raspberry Pi, como la detección de movimiento, alarma sonora, captura de imágenes y video, era necesario disponer de ciertos componentes eletrónicos.

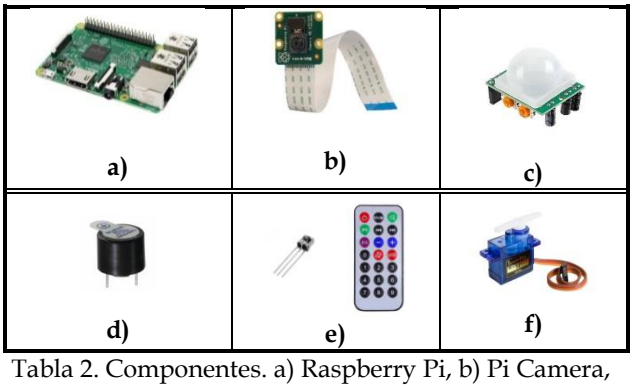

Tabla 2. Componentes. a) Raspberry Pi, b) Pi Camera, c) Sensor Pir, d) Buzzer, e) IR Remote, f) Servo Motor

- **Raspberry Pi 3**. Es un ordenador de placa reducida de bajo coste.
- **Pi Camera**. Es una cámara de 5MP capaz de capturar vídeo a 1080p (1920x1080) y también imágenes una vez conectada a la Raspberry Pi.

- **Sensor PIR (HC-SR501).** Es un elemento que detecta cambios en la radiación infrarroja que percibe y dispara una alarma al percibirlo. Detectará presencias en el sistema.
- **Buzzer**. Es un componente capaz de transformar la electricidad en sonido. Cuando el sistema detecte un movimiento el buzzer emitirá unos pitidos.
- **IR Remote**. Es un módulo receptor infrarrojos que permite controlar dispositivos con un control remoto. Permite al usuario activar o desactivar el sistema de videovigilancia sin necesidad de tener el Smartphone.
- **Servo Motor**. Es un motor electrónico de baja inercia al que se le puede controlar tanto la velocidad de giro como la posición dentro de su rango de operación. Permite rotar la posición de la Pi Camera según el usuario desee.

Tal y como se ha comentado anteriormente, el sistema de videovigilancia dispondrá de una plataforma web para poder almacenar todo aquello que el sistema haya detectado. Para que esta información sea almacenada de forma segura, se añade una segunda Raspberry Pi. El uso de la segunda Raspberry Pi no solo es para crear un servidor de página web (ya que se podría hacer perfectamente en la Raspberry Pi que ejerce de sistema de videovigilancia) sino que permite al usuario, en el caso de que se extraviara el sistema de videovigilancia, no perder aquellas imágenes y videos que el sistema haya capturado al detectar alguna presencia ya que se almacenaría en el Raspberry Pi que ejerce de servidor web y se podría acceder a la información en cualquier momento.

7.1.2. Arquitectura hardware

La arquitectura hardware a nivel general del proyecto se compone principalmente de dos Raspberry's Pi, una para gestionar todos los componentes electrónicos y la otra para ejecutar el servidor web. Las dos Raspberry's Pi estarán conectadas entre ellas de manera inalámbrica, a través de protocolos de comunicación. En la Figura 2 se muestra la arquitectura del proyecto.

Con respecto a la Raspberry que se encarga de gestionar los diferentes componentes electrónicos, una vez hecha la lista de los componentes a utilizar en el proyecto había que leer el *datasheet* (ficha técnica) de cada componente para diseñar la arquitectura de conexiones. Para entender mejor esta arquitectura, en la Figura 1 se muestra el circuito esquemático de los componentes eletrónicos conectados a la Raspberry Pi.

- **IR Remote**: este receptor de infrarrojos tiene tres señales. La señal de alimentación debe alimentarse a 3,3V por motivos de seguridad, ya que un voltaje mayor podría dañar el componente o incluso la Raspberry Pi. El pin 4 de la GPIO se encargará de controlar las señales infrarrojas, y por último, la señal GND que ira conectada a uno de los pin de GND de la Raspberry Pi.
- **Buzzer**: este componente se compone de dos señales. Para poder alimentarlo (que es cuando emitira los zumbidos) lo controlaremos a partir del pin 10 de la Raspberry. Es decir, cuando el pin 10 se active, dejando pasar una corriente de 3,3V, el Buzzer efectuara un zumbido. La señal de tierra se conecta a uno de los pin GND de la Raspberry Pi.
- **Sensor PIR**: este sensor de detección de movimiento tiene tres señales. Al igual que el IR Remote, para evitar daños se ha de alimentar a 3,3V. El pin 21 de la Raspberry Pi se encargará de detectar si el sensor detecta o no una presencia y alertará al sistema. Para la señal de tierra se debe de conectar a un pin GND de la Raspberry Pi.
- **Servo Motor**: este componente, que permite control la posición del eje para poder rotar la posición de la cámara, contiene tres señales. El Servo Motor necesita ser alimentado por 5V, por lo tanto, esta señal ira conectada al pin de la Raspberry Pi que nos proporciona 5V. La señal que recibirá los pulsos y hará que el Servo Motor cambie la posición del eje estará conectada al pin 18 de la Raspberry Pi. La señal GND del componente, al igual que el resto de módulos, estará conectado a un pin GND de la Raspberry Pi.
- **PiCamara:** este componente, que ejerce de papel de cámara, está conectado mediante un bus de cinta al conector especial que hay junto al conector Ethernet de la Raspberry Pi.

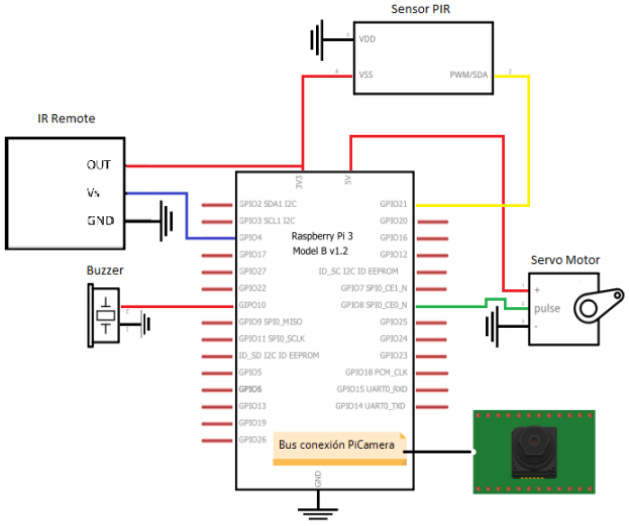

Figura 1. Esquema electrónico

7.2 Software

7.2.1. Arquitectura software

Para desarrollar una arquitectura de software que me proporcionase un marco definido y claro para interactuar

con el código fuente software necesitaba conocer el diseño de más alto nivel de la estructura del sistema. Esta estructura del sistema se componía de tres sujetos:

- **Bot**. Sujeto que se encarga de enviar las peticiones solicitadas por el usuario al servidor bot.
- **Servidor bot**. Sujeto que ejecuta el Bot y gestiona las peticiones recibidas. Este servidor Bot se ejecuta en la Raspberry Pi que gestiona los componentes electrónicos.
- **Servidor web**. Sujeto que crea el servidor de la página web.

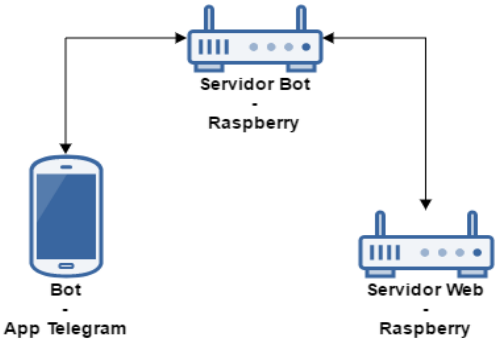

Figura 2. Sujetos del sistema

La Raspberry Pi, que gestiona los componentes electrónicos, ejecuta el servidor bot utilizando la API Telegram. En ese momento, el servidor Bot se sincroniza con los servidores Telegram identificándose mediante un *token*. Este token es una cadena de valores que comparte tanto el Bot del Smartphone como el servidor Bot. Esto permite que las peticiones enviadas por parte del Bot desde el Smartphone las reciba su propio servidor Bot.

Así pues, en el momento en el que desde el Bot envié una petición, está será recibida por el servidor Telegram y redirigida por él hacia su respectivo servidor Bot gracias al token que comparten.

Una vez obtenido tanto el catálgo de requisitos como el diseño de la arquitectura hardware y la arquitectura de comunicación entre los diferentes sujetos, se ha pasado al análisis y diseño, aplicando una Ingeniería del Software Orientado a Objetos, siguiendo el modelo UML (Unified Modeling Language). En el Apéndice 3 se detalla el modelo UML de este proyecto.

7.2.2. Programas, aplicaciones y librerías

Conociendo las necesidades del proyecto, se seleccionó los siguientes programas y aplicaciones:

- Raspbian (Distribución SO Linux Raspberry)
- Aplicación Telegram
- API Telegram
- Pip (Administrador de paquetes software)
- Sublime Text 3 (Editor de texto)
- MobaXterm (cmd)
- NinjaMock (Diseño de wireframes)
- BitBucket (Servicio de alojamiento)

7.2.3. Lenguajes informáticos

Para seleccionar los lenguajes informáticos que iban a ser utilizados, necesitaba saber cual era el lenguaje de programación adecuado para trabajar con la Raspberry Pi y los lenguajes a utilizar para desarrollar una plataforma web.

Así pues, se utilizaron los siguientes lenguajes de programación:

- Python
- HTML y CSS
- Ajax

Para implementar las funcionalidades del sistema se utilizó el lenguaje Python ya que se trata del lenguaje de programación "estándar" para Raspberry Pi. Es un lenguaje bastante potente y con muchas librerías que ayuda a realizar cualquier cosa, en este caso, el control de los *pines* de la Raspberry Pi para poder hacer uso de los diferentes componentes electrónicos conectados a ella. Para desarrollar la aplicación web se utilizó el framework *Flask* [8]. Este framework está basado en Python y permite crear aplicaciones web. También fue utilizado para el *back-end* de la plataforma. Para el *front-end*, HTML y CSS. Para que la web fuese iterativa se ha utilizado el lenguaje de programación Ajax, que nos ha permitido controlar las rotaciones de la cámara del sistema mediante unos botones en la web.

7.2.4. Bot Telegram

Tal y como se ha comentado anteriormente, el intermediario entre el sistema de videovigilancia y el usuario es el servicio de mensajería Telegram. El usuario desde esta aplicación puede gestionar el sistema de videovigilancia. Pero para que la aplicación Telegram se entendiese con la Raspberry Pi, era necesario el uso de un *Bot Telegram* [9].

Un *Bot* [10] es un software que imita un comportamiento humano, como Siri o Cortana. Telegram permite a sus usuarios crear Bots y comunicarse con ellos mediante los *chatbots* [11]. Con el *chatbot*, el bot será capaz de simular una conversación con el usuario del sistema de videovigilancia desde Telegram para gestionar el sistema.

8 IMPLEMENTACIÓN DEL SISTEMA

En esta sección se explicará cómo se ha desarrollado las características del sistema de videovigilancia y el desarrollo de la plataforma web separando cada uno de los bloques.

8.1 Sistema de seguridad

El primer paso fue implementar la comunicación entre la Raspberry Pi y la aplicación Telegram. Teniendo claro la arquitectura de comunicación descrita en la sección anterior, se desarrolló el servidor Bot en la Raspberry Pi para hacer uso del Bot Telegram y darle vida en el Smartphone. Esto fue posible utilizando la API Telegram.

Una vez ejecutado el servidor Bot en la Raspberry Pi, se desarrolló un sistema para que fuese realmente sencillo enviar peticiones desde el chatbot al sistema de videovigi-

lancia. Para ello, se optó por establecer un menú de opciones intuitivo en el chatbot. El menú se formaba de cuatro opciones:

- **Activar sistema.** Pone en marcha el sistema de videovigilancia. En el momento que detecté alguna presencia el usuario será notificado en su Smartphone desde la aplicación Telegram. A demás, se le adjuntará una imagen y un corto video para obtener más información de lo sucedido.
- **Desactivar sistema**. Deja el sistema totalmente vulnerable a cualquier presencia ya que no detectaría ningún tipo de presencia.
- **Captura instantánea.** Permite al usuario solicitar una captura de imagen en ese preciso momento.
- **Video streaming**. Generará al usuario un sitio web para poder visualizar de forma segura lo que esté sucediendo en streaming.

8.1.1. Menú de opciones

Para el desarrollo del menú de opciones, la API Telegram me permetía implementar diferentes *keyboards* [12]. Opté por un menú vertical, en el cual se mostraba en el chatbot 4 opciones y con un solo clic en una de ellas se enviaba la petición.

8.1.2. Captura instantanea

Finalizado el menú de opciones, daba paso a las implementaciones del comportamiento esperando al seleccionar una de las opciones del menú.

La primera opción del menú que se implementó fue la opción *Captura Instantánea.* Empezar por la opción *Activar Sistema* suponía implementar todos los componentes electrónicos (sensor PIR, buzzer…) a la vez. Para la opción *Captura Instantánea* era necesario el uso de la Pi Camera ya que nos permitirá realizar una captura de imagen y seguidamente enviarla al usuario. Utilizando la librería *picamera* me permitió realizar capturas y establecer una configuración a la Pi Camera, estableciendo una resolución de 640x480. Esta misma configuración era útil para el *video streaming*.

8.1.3. Video streaming

Finalizada la opción *Captura Instantánea* se le dio paso a la opción *Video Streaming* ya que había implementado parte del módulo Pi Camera que era útil porque en esta opción seguía trabajando con el componente. Esta opción, una vez solicitada, el usuario recibía una url y contraseña para acceder al streaming.

Para poder implementar esta opción, seguí utilizando la librería *picamera* y utilicé el framework Flask. Tal y como se ha comentado anteriormente, este framework permite realizar aplicaciones web. Se utilizó el framework para crear una aplicación web en la que el usuario pudiese visualizar en streaming el sistema haciendo uso de la Pi Camera.

A demás, se implementó un sistema de autentificación *OTP* [13] para cada petición streaming. Una OTP es una contraseña valida solo para una autentificación. Esto se añadió para darle una capa de seguridad y ser más resistentes frente ataques de *fuerza bruta.* Para complicar todavía los ataques, la url que recibe el usuario en cada petición sería una url dinámica.

Un componente más que se ha utilizado en esta opción ha sido el Servo motor. El servo motor controla la velocidad de giro para poder rotar la Pi Camera mediante la modulación por ancho de pulsos, es decir, *PWM* [14]. El PWM es una técnica que consiste en variar el ancho de puslo de una señal de voltaje cuadrada con el objetivo de controlar la cantidad de potencia administrada en el servor motor. Para que el usuario pueda rotar la Pi Camera, se implementó unos botones de control desarrollados en la aplicación web del streaming.

8.1.5. Activar/Desactivar sistema

Una vez implementada la Pi Camera, era posible implementar la opción *Activar sistema* ya que disponer de la Pi Camera implementada era un requisito previo.

La idea para esta opción era la siguiente: cuando el usuario activase la opción *Activar sistema*, el sistema de seguridad se activaba para detectar cualquier movimiento. Con la ayuda del Sensor PIR esto sería posible.

Para el control del Sensor PIR se utilizó los pines de la Raspbery Pi que van conectados al componente. El sensor PIR, al detectar cambios de radiacción infrarroja, le envía una señal a la Raspberry Pi. Una vez detectada la presencia, con la Pi Camera se capturaría una imagen y un video de periodo corto para ser enviado al usuario.

Justo después de este proceso, utilizando el componente Buzzer, se enviaría unos pitidos para generar una alarma sonora presenciando alerta de intruso. Esto ocurre cuando la Raspberry Pi le da corriente al componente.

Para que tanto la captura de imagen como video fuesen almacenadas de forma segura, se pretendía que en el momento en el que el usuario recibía esa información a través de la aplicación Telegram, se enviase al servidor web para poder visualizar en la plataforma web el contenido.

Al no tener implementada la plataforma web, el requisito de enviar la información recibida por el sistema hacia el servidor web fue necesario implementarlo más adelante, una vez finalizada la plataforma web.

La última opción por implementar, *Desactivar sistema*, como se ha comentado anteriormente, era poner el sistema vulnerable a cualquier presencia.

Estas dos opciones, tanto *Activar sistema* como *Desactivar* sistema es posible gestionarlo desde un control remoto. Para ello, se utiliza el IR Remote, que tal y como he comentado anteriormente, es un receptor infrarrojos que permite controlar dispositivos con un control remoto. Fue necesario configurar, tanto el control remoto como el IR Remote con la Raspberry Pi. Para el control remoto se identificó los botones que iban a ser utilizados para que al presionarlos el IR Remote supiese que valor contenía, y así saber si había que activar o desactivar el sistema.

Para entender mejor la opción *Activar sistema*, en la Figura 3 se muestra un diagrama de flujo.

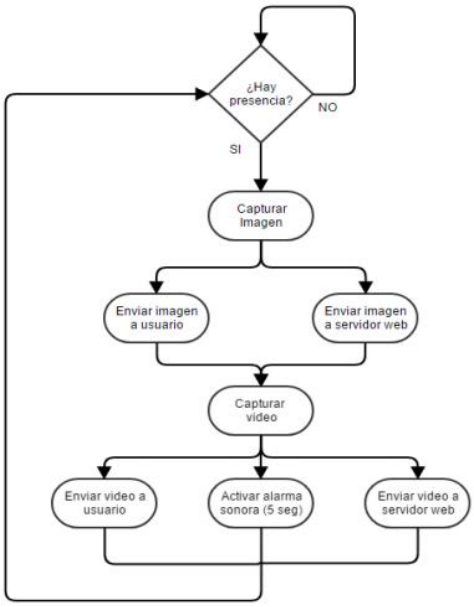

Figura 3. Diagrama de flujo *Activar Sistema*

8.1.6. Diagrama de secuencia del sistema

A continuación, en la Figura 4, se muestra un diagrama de secuencia para entender la interacción entre los objetos del sistema.

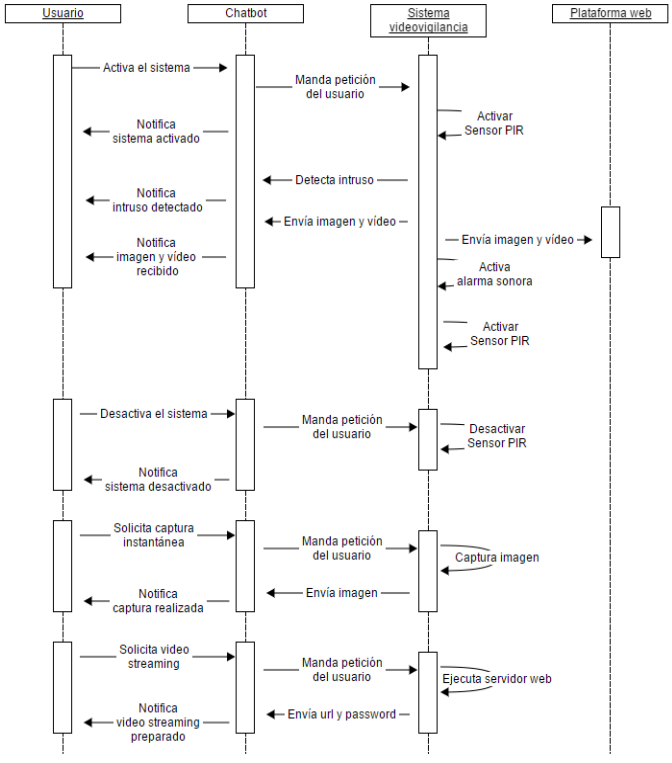

Figura 4. Diagrama de flujo *Activar Sistema*

8.2 Plataforma web

En este punto, se hace referencia al desarrollo de la plataforma web en la segunda Raspberry Pi, la cual ejecuta el servidor para crear la plataforma web.

En la plataforma web el usuario podrá acceder y visualizar las imágenes y videos que el sistema de videovigilancia haya recogido al detectar una presencia. A parte, dispone de una sección para poder visualizar un video streaming transmitido por la Pi Camera.

8.2.1. Diseño plataforma web

El primer paso para desarrollar la plataforma web fue realizar su diseño. La idea era desarrollar una plataforma web muy intuitiva y sencilla para el usuario. Para ello se realizó tres *wireframe*, una para cada página: pagina inicio, página historial y pagina streaming.

La *página inicio* es la página de acceso a la web. En ella, el usuario tiene que identificarse para acceder. La *página historial* es la página que contiene todas las imágenes y videos recogidas por el sistema de videovigilancia al detectar una presencia. La *página streaming* permite al usuario visualizar en streaming todo aquello que visualice la Pi Camera del sistema.

8.2.2. Página Inicio

Una vez diseñada la plataforma web, daba paso a la implementación. La página Inicio, es la página principal, aquella en la que el usuario ha de introducir unos datos para acceder a toda la información. Para este control de acceso se utilizó el framework *Flask* para el desarrollo de un login.

8.2.2. Página Historial

El siguiente paso fue implementar la página Historial. Anteriormente, se ha comentado que la plataforma web recibiría las imágenes y videos que el sistema recoja al detectar una presencia. Para ello, era necesario comunicar el sistema de videovigilancia con el servidor web para poder transferir de una Raspberry a otra toda la información recogida (imágenes y videos). Esta información era enviada a través del protocolo SSH. El problema, era que en cada envío era necesaria una autentificación para acceder y enviar la información. Para permitir la transferencia de la información desde la Raspberry Pi, que ejerce como sistema de videovigilacia, a la Raspberry Pi, que actúa como servidor web, sin tener que enviar la password para acceder, se implementó una autentificación basada en *Public Key* [15]. En el Apéndice 4 se detalla este tipo de autentificación.

8.2.2. Página Streaming

Por último, era necesario enviar el flujo de datos que transmitita la Pi Camera desde la Raspberry Pi del sistema de videovigilancia a la Raspberry Pi de servidor web para que el usuario pudiera realizar un video streaming en la página Streaming. Al trabajar con la transferencia de datos entre dos programas distintos, era necesario utilizar *sockets* [16]. A través de los *sockets* era posible que las dos Raspberry's Pi, el sistema de videovigilancia y el servidor web, pudiesen intercambiar cualquier flujo de datos de manera fiable y ordenada, en este caso, los fotogramas de la Pi Camera. En el Apéndice 5 se muestra el diagrama de flujo entre las dos Raspberry's para entender el desarrollo con sockets.

8.3 Conexión desde Internet

La parte más atractiva de este proyecto es la posibilidad de conectarte al sistema de videovigilancia desde fuera de casa.

Para ello, era necesario configurar una IP local fija para cada una de las dos Raspberry's Pi. Seguidamente, había que redireccionar los puertos del router para acceder a las Raspberry's Pi de la red local desde internet. Este proceso consistía en abrir un puerto en el router y direccionarlo a la Raspberry Pi con el puerto correspondiente.

Una vez hecho esto, era posible gestionar el acceso al video streaming desde fuera de casa.

9 TEST DE VALIDACIÓN

Para comprobar determinados aspectos del software desarrollado se han ejecutado diferentes pruebas de software cuyo objetivo ha sido proporcionar información objetiva e independiente sobre la calidad del proyecto. A continuación se describe cada nivel de prueba:

- **Pruebas unitarias**. Este tipo de prueba consiste en probar las unidades del software al más bajo nivel, el de programación. Estas pruebas han permitido comprobar si los métodos de las clases diseñadas funcionaban correctamente de forma aislada.
- **Pruebas de integración**. Este tipo de prueba consiste en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta. Estas pruebas han permitido verificar que los diferentes componentes electrónicos funcionaban correctamente actuando en conjunto.
- **Pruebas de sistema**. Este tipo de prueba consiste en probar el sistema software completo e integrado, para verificar desde el punto de vista de requisitos que el sistema funciona tal y como se detalló.

En Tabla 3 se muestran algunas de las pruebas realizadas. Se puede observar que a medida que se añadían nuevas características al proyecto éstas eran sometidas a pruebas para verificar sus funcionalidades, pero estas pruebas eran en entornos muy controlados.

Para poder verificar algunos aspectos del proyecto de forma real, se ha permitido a un grupo reducido de personas el acceso al proyecto. Las conclusiones extraídas por parte de los usuarios son las siguientes:

- Respecto a la interface, compatibilidad y uso del menú de opciones en el chatbot han estado correctas y en la línea esperada, es una interficie fácil de utilizar y adáptale a diferentes dispositivos ya que se trabaja con la aplicación Telegram.
- En referencia a las peticiones Captura de imagen y Video Streaming han cumplido con las necesidades de los usuarios. El comportamiento del sistema al detectar un intruso es el esperado.
- Respecto a la información proporcionada por la plataforma web y la petición de video streaming se realizan correctamente.

Los comentarios por parte de los usuarios hacen hincapié en el funcionamiento correcto de las funcionalidades.

Tipo de prueba	Módulo	Descripción	Pass/Fail
Prueba unitaria	Pi Camera	Capturar imagen	**Pass**
Prueba unitaria	Pi Camera	Capturar video	**Pass**
Prueba unitaria	Sensor PIR	Detectar movimiento	**Pass**
Prueba unitaria	Buzzer	Emitir zumbido	**Pass**
Prueba unitaria	IR Remote	Recibir señales	**Pass**
Prueba unitaria	IR Remote	Interpretar señales	**Pass**
Prueba unitaria	Servo motor	Rotar 45º	**Pass**
Prueba integración	Pi Camera, /Telegram	Video streaming en Smartphone	**Pass**
Prueba integración	Pi Camera/ Sensor Pir	Imagen/ video al detectar presencia	**Pass**
Prueba unitaria	Web	Login	**Pass**
Prueba integración	Pi Camera/ Web	Video streaming en web	**Pass**
Prueba Sistema	*	Activar sistema	**Pass**

Tabla 3. Niveles de prueba software

10 RESULTADOS

A continuación se muestran los resultados de la implementación del sistema de videovigilancia, validando cada uno de los objetivos expuestos en la sección *3. Objetivos* de este informe.

El mayor problema ha sido la baja señal que había en toda la casa. Para tener una idea de la conexión utilicé un medidor online para comprobar la velocidad. Los datos fueron los siguientes:

- Ping: 109 ms
- Descarga: 1.69Mbps
- Carga: 0.07Mbps

Este problema es crucial para aquellos usuarios que tengan una conexión débil en casa.

10.1. Sistema de seguridad

10.2.1. Opción Activar sistema

El tiempo transcurrido desde que se detecta una presencia hasta recibir la imagen y el vídeo del intruso en la aplicación Telegram, es aproximadamente de 60 segundos.

10.2.2. Opción Capturar instantánea

Al solicitar esta opción, el usuario recibe la imagen en aproximadamente 15 segundos. A pesar de que la imagen tenga una resolución de 640x480 la baja conexión en casa impide recibirla en un periodo de tiempo más corto.

10.2.3 Opción Video streaming

Para esta opción se ha ejecutado de dos formas, solicitando el video streaming con wifi y con 4G.

Para el video streaming con wifi, a pesar de tener una conexión muy débil, se ha comprobado que se realiza un streaming bastante fluido. Respecto al streaming en 4G se puede apreciar cierto lag dando a lugar a un straming de baja fluidez.

10.2. Plataforma Web

10.2.1 Implementación de la página Streaming

En este punto es donde más ha sufrido el sistema. La transferencia del flujo de datos que retransmitía la Pi Camera, desde el sistema de videovigilancia a la plataforma web, se convierte en un streaming de muy baja calidad. Se puede apreciar un retardo excesivo provocado por la poca potencia de procesamiento, tanto en el cliente (sistema de videovigilancia) y el servidor (plataforma web), con el que se establece la comunicación.

10.3. Demostración del sistema del proyecto

A continuación se comparten unos links para demostrar el funcionamiento de las diferentes características del sistema de videovigilancia.

- Imágenes – Sistema de videovigilancia:
En el Apéndice 2 se muestra el diseño final del sistema desarrollado.

- Vídeo 1 – Activar sistema:
https://www.youtube.com/watch?v=tFixAZ5UNQ8

- Vídeo 2 – Captura instantánea:
https://www.youtube.com/watch?v=A7NdLQPepS0

- Vídeo 3 – Video streaming mediante wifi:
https://www.youtube.com/watch?v=bMJBCfA5EYU

- Vídeo 4 – Video streaming mediante 4G:
https://www.youtube.com/watch?v=qclLm8-YKn8

- Vídeo 5 – Funcionamiento Servo motor:
https://www.youtube.com/watch?v=YTHMPFxptoI

- Vídeo 6 – Plataforma web:
https://www.youtube.com/watch?v=N3CNxeuYcEA

11 Conclusión y trabajo futuro

Tal y como se ha podido comprobar a lo largo de este artículo, se han cumplido los objetivos establecidos para este proyecto.

Para poder alcanzar el cumplimiento de los objetivos se ha ido haciendo un seguimiento de procesos de la ingeniería del software que ha permitido desarrollar el proyecto de la mejor manera posible.

El resultado es muy satisfactorio ya que se ha conseguido desarrollar un sistema de videovigilancia que funciona y se puede demostrar.

A nivel personal, este proyecto ha resultado ser un desafío en muchos sentidos debido a la utilización de tecnologías desconocidas como la API Telegram, el framework Flask o el uso de diferentes componentes electrónicos, entre otros.

Este proyecto me ha aportado una profunda formación y aprendizaje desde el primer minuto, arrastrándome a un gran interés e ideas para un futuro, en el desarrollo de proyectos con la Raspberry Pi.

11.1 Problemas encontrados

El principal inconveniente del proyecto es la necesidad de tener que estar conectado a la corriente para su funcionamiento, ya que en caso de pérdida de energía el sistema de videovigilancia deja de ser útil.

Otro inconveniente es la poca efectividad del sistema en lugares oscuros ya que las imágenes, videos y el streaming son transmitidos por la Pi Camera sin ningún tipo de luz externa.

Para velocidades de conexiones lentas, tal y como ha sucedido en mi caso, genera un periodo de tiempo excesivamente grande al solicitar imágenes o recibir las pruebas al detectar algún intruso, y mayormente encontrar lag a la hora de realizar streaming.

11.2 Futuras lineas de mejora

Las posibilidades de mejorar este proyecto son ilimitadas ya que la estructura del proyecto es bastante flexible para poder trabajar con nuevos componentes electrónicos.

La utilización de un nuevo servo motor permitiría poder rotar la cámara tanto vertical como horizontalmente y tener mejor visualización.

El control de acceso a la vivienda por parte del sistema sería una gran mejora en el proyecto ya que permitiría al usuario gestionar el acceso al hogar desde cualquier punto de la casa, sin necesidad de tener que utilizar un videoportero.

Cambiar la Pi Camera por una Pi Camera con visión nocturna solucionaría el problema de la poca efectividad que tiene el sistema en la oscuridad.

Para resolver el problema con la energía que necesita el sistema para estar en funcionamiento, en caso de apagón o corte de luz, el sistema seguiría en funcionamiento con un sistema de alimentación interrumpida (SAI). Se trata de un dispositivo que gracias a sus baterías puede pro-

porcionar energía eléctrica a todos los dispositivos que tenga conectados, en este caso, las dos Raspberry's Pi.

Una gran mejora de cara a implementar en el proyecto, es la utilización de la liberia *OpenCV*. Se trata de una biblioteca libre de visión artificial. Utilizar OpenCV en el proyecto abre un mundo infinito para aumentar las características del proyecto. Un ejemplo sería el reconocimiento facial, esto permetiría evitar que nos detecte el sistema como intruso ya que el sistema nos reconocería y no haría saltar la alarma.

AGRADECIMIENTOS

Agradecer al tutor de este trabajo, Fernando Vilariño, su ayuda y consejos en la elaboración de documentos y el enfoque del trabajo que se ha ido desarrollando a lo largo de estos meses de trabajo.

Agradecer a mi familia por todo el esfuerzo durante mi etapa como universitario, y mi pareja el soporte incondicional en los momentos más difíciles.

BIBLIOGRAFÍA

[1] Dario El País "Sistemas de seguridad para el hogar" [En linea] Disponible en: http://economia.elpais.com/economia/2014/02/27/vivienda/1393503804_290817.html

[2] Raspberry Pi "What is a Raspberry Pi?" [En linea] Disponible en: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

[3] Telegram "What is Telegram?" [En linea] Disponible en: https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here

[4] Ignacio Bartolomé "Sistema de videovigilancia low-cost" [En linea] Disponible en: http://eprints.ucm.es/31300/1/Memoria%20TFG%20SecBerry-Sistema%20de%20videovigilancia%20lowcost%20sin%20Autorizaci%C3%B3n.pdf

[5] Laura Quilis "Software de videovigilància per a Raspberry Pi, amb accés segur per a control d'un actuador" [En linea] Disponible en: https://riunet.upv.es/bitstream/handle/10251/54472/tfg%20laura%20quilis.pdf?sequence=9

[6] Agile Modeling "Feature Driven Development". [En linea] Disponible en: http://agilemodeling.com/essays/fdd.htm

[7] Agile Management Practices "Agile Development Sprint Planning". [En linea] Disponible en: https://www.versionone.com/agile-101/agile-management-practices/agile-development-iteration-planning/

[8] Flask "Flask, Web development". [En linea] Disponible en: http://flask.pocoo.org/

[9] Computer Hoy "¿Qúe son los bots de Telegram y como usarlos?". [En linea] Disponible en: http://computerhoy.com/paso-a-paso/apps/que-son-bots-telegram-como-usarlos-43505

[10] CNET "¿Qué es un bot?". [En linea] Disponible en: https://www.cnet.com/es/como-se-hace/que-es-un-bot/

[11] Diario elEconomista "¿Qué son exactamente los chatbots y para que sirven?". [En linea] Disponible en: http://www.eleconomista.es/tecnologia/noticias/7488529/04/16/Que-son-exactamente-los-chatbots-y-para-que-sirven.html

[12] API Telegram Manual "Bots: An introduction for developers" [En linea] Disponible: https://core.telegram.org/bots

[13] Search Security "One-time password (OTP)". [En linea] Disponible en: http://searchsecurity.techtarget.com/definition/one-time-password-OTP

[14] Nathaniel Pinckney "Pulse-width modulation for microcontroller servo control" [En linea] Disponible: http://tech-uofm.info/fall_2013/TECH4234/PWM.pdf

[15] David McNett "Using SSH Public Key Authentication". [En linea] Disponible en: https://macnugget.org/projects/publickeys/

[16] Tutorials Point "What is a Socket?" [En linea] Disponible en: https://www.tutorialspoint.com/unix_sockets/what_is_socket.htm

APÉNDICE

A1. DIAGRAMA DE GANNT

En este apéndice se puede observar los sprints programados desde el inicio del proyecto en un diagrama de Gantt.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Trabajo	Nombres de los recursos	% trabajo completado	Costo	Costo real
1		**Sistema de videovigilancia a través de una Raspberry Pi**	**79 días**	**mar 24/01/17**	**vie 12/05/17**	**180 horas**		**100%**	**4.125,00 €**	**4.125,00 €**
2		**Sprint 0**	**9 días**	**mar 24/01/17**	**vie 03/02/17**	**17 horas**		**100%**	**365,00 €**	**365,00 €**
3		Definición y diseño de los objetivos del proyecto	3 días	mar 24/01/17	jue 26/01/17	5 horas	Ingeniero de Requisitos	100%	125,00 €	125,00 €
4		Instalación y configuración de la Raspberry PI	1 día	vie 27/01/17	vie 27/01/17	2 horas	Programador	100%	40,00 €	40,00 €
5		Recopilación de fuentes bibliograficas	5 días	lun 30/01/17	vie 03/02/17	10 horas	Programador	100%	200,00 €	200,00 €
6		**Sprint 1**	**5 días**	**lun 06/02/17**	**vie 10/02/17**	**11 horas**		**100%**	**220,00 €**	**220,00 €**
7		Instalacion y configuracion del Bot Telegram	1 día	lun 06/02/17	lun 06/02/17	1 hora	Programador	100%	20,00 €	20,00 €
8		Configurar conexión Raspberry Pi - Bot Telegram	2 días	lun 06/02/17	mar 07/02/17	4 horas	Programador	100%	80,00 €	80,00 €
9		Introducir keyboard en la aplicación Telegram	2 días	mié 08/02/17	jue 09/02/17	6 horas	Programador	100%	120,00 €	120,00 €
10		**Sprint 2**	**15 días**	**lun 13/02/17**	**vie 03/03/17**	**32 horas**		**100%**	**1.200,00 €**	**1.200,00 €**
11		Implementar petición "captura instantanea"	2 días	lun 13/02/17	mar 14/02/17	2 horas	Programador	100%	400,00 €	400,00 €
12		Implementar petición "video streaming" en "local"	5 días	mié 15/02/17	mar 21/02/17	20 horas	Programador	100%	400,00 €	400,00 €
13		Implementar petición "video streaming" en "remoto"	3 días	mié 22/02/17	vie 24/02/17	10 horas	Programador	100%	200,00 €	200,00 €
14		Pruebas y resolución de problemas	5 días	lun 27/02/17	vie 03/03/17	10 horas	Tester	100%	200,00 €	200,00 €
15		**Sprint 3**	**5 días**	**lun 06/03/17**	**vie 10/03/17**	**12 horas**		**100%**	**240,00 €**	**240,00 €**
16		Implementar modulo HC-SR501 (Sensor PIR)	3 días	lun 06/03/17	mié 08/03/17	7 horas	Programador	100%	140,00 €	140,00 €
17		Pruebas y resolución de problemas	2 días	jue 09/03/17	vie 10/03/17	5 horas	Tester	100%	140,00 €	140,00 €
18		**Sprint 4**	**5 días**	**lun 13/03/17**	**vie 17/03/17**	**12 horas**		**100%**	**240,00 €**	**240,00 €**
19		Implementar modulo Buzzer	3 días	lun 13/03/17	mié 15/03/17	7 horas	Programador	100%	140,00 €	140,00 €
20		Pruebas y resolución de problemas	2 días	jue 16/03/17	vie 17/03/17	5 horas	Tester	100%	100,00 €	100,00 €
21		**Sprint 5**	**15 días**	**lun 20/03/17**	**vie 07/04/17**	**36 horas**		**100%**	**720,00 €**	**720,00 €**
22		Analaisi de nuevos componenetes	5 días	lun 20/03/17	vie 24/03/17	12 horas	Programador	100%	240,00 €	240,00 €
23		Implementación de nuevos componenetes	10 días	lun 27/03/17	vie 07/04/17	24 horas	Programador	100%	480,00 €	480,00 €
24		**Sprint 6**	**5 días**	**lun 10/04/17**	**vie 14/04/17**	**12 horas**		**100%**	**180,00 €**	**180,00 €**
25		Diseño de la plataforma web	3 días	lun 10/04/17	mié 12/04/17	7 horas	Diseñador	100%	105,00 €	105,00 €
26		Diseño base de datos	2 días	jue 13/04/17	vie 14/04/17	5 horas	Diseñador	100%	75,00 €	75,00 €
27		**Sprint 7**	**10 días**	**lun 17/04/17**	**vie 28/04/17**	**24 horas**		**100%**	**480,00 €**	**480,00 €**
28		Implementación Página Inicial	3 días	lun 17/04/17	mié 19/04/17	7 horas	Programador	100%	140,00 €	140,00 €
29		Implementación Loggin	2 días	jue 20/04/17	vie 21/04/17	5 horas	Programador	100%	100,00 €	100,00 €
30		Implementación Página Historial	3 días	lun 24/04/17	mié 26/04/17	7 horas	Programador	100%	140,00 €	140,00 €
31		Pruebas y resolución de problemas	2 días	jue 27/04/17	vie 28/04/17	5 horas	Tester	100%	100,00 €	100,00 €
32		**Sprint 8**	**10 días**	**lun 01/05/17**	**vie 12/05/17**	**24 horas**		**100%**	**480,00 €**	**480,00 €**
33		Implementar comunicación	5 días	lun 01/05/17	vie 05/05/17	12 horas	Programador	100%	240,00 €	240,00 €
34		Pruebas y resolución de problemas	5 días	lun 08/05/17	vie 12/05/17	12 horas	Tester	100%	240,00 €	240,00 €

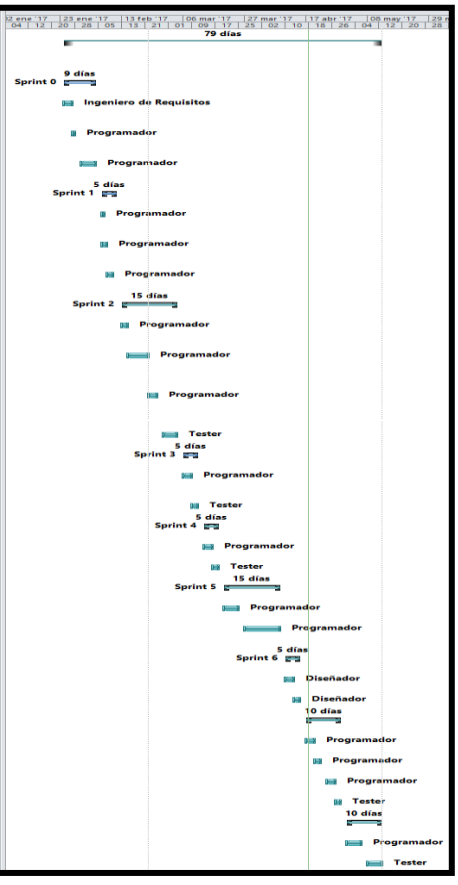

Figura 5. Diagrama de Gantt

A2. DISEÑO FINAL DEL SISTEMA

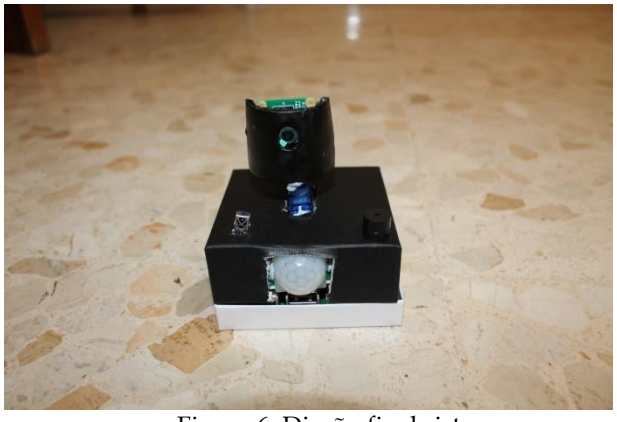

Figura 6. Diseño final sistema

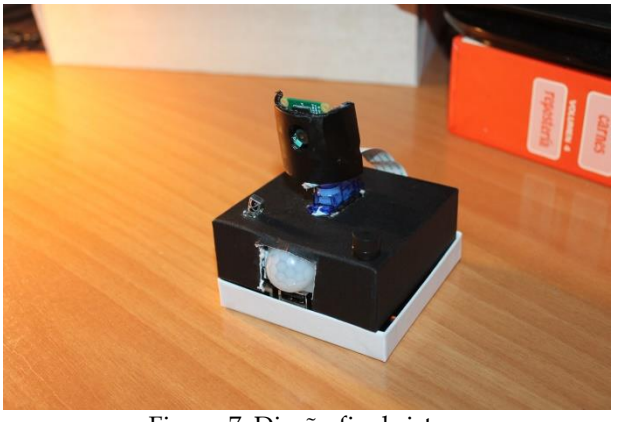

Figura 7. Diseño final sistema

A3. DIAGRAMA UML

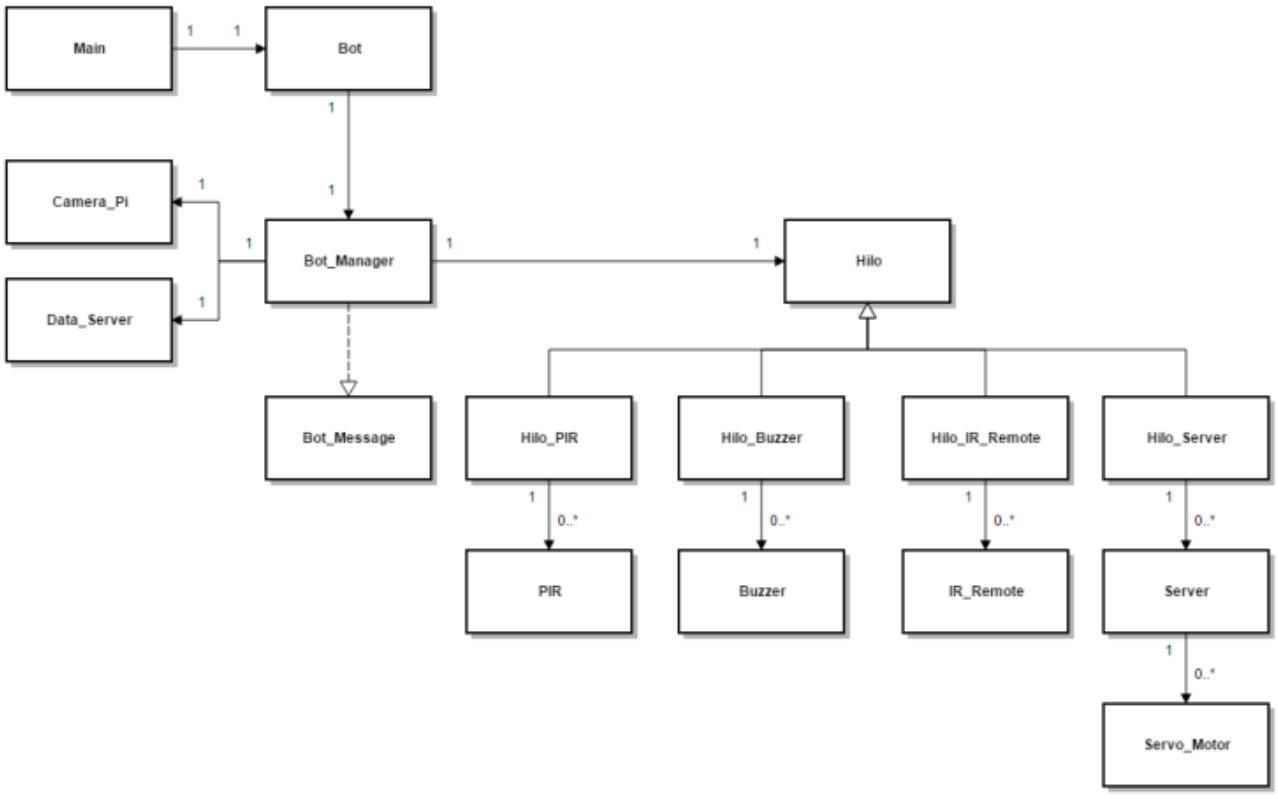

Figura 8. Diagrama de Gantt

Para esta arquitectura de software del sistema de videovigilancia cada clase tiene sus definiciones de las propiedades y comportamiento que logran cumplir dicha funcionalidad especial.

Para poner en marcha el sistema, la **clase Main** es la encargada de que se ejecute. Es el punto de entrada que especifica dónde debe de comenzar la ejecución del programa.

La **clase Bot** es quien ejecuta el *chatbot* para que el usuario pueda interactuar con el sistema.

La **clase Bot_Manager** gestiona las peticiones enviadas por el usuario.

La **clase Camera_Pi** es la clase que gestiona las funcionalidades principales de la Pi Camara: capturar una imagen o captar en tiempo real a modo de video.

La **clase Data_Server** es la que añade una capa de seguridad al solicitar una url para acceder al video streaming. Generará una url y password aleatoria por cada petición del usuario al video streaming. Esto es importante de cara a evitar el acceso de cualquier persona que escuche la red.

Para la gestión de la respuesta por parte del Bot, como el envió de la captura de imagen, la url para el video en streaming o las alertas de presencia, entre otras, la gestionará la **clase Bot_Message**. Se encargará de enviar la respuesta al usuario.

La **clase Hilo** es la que gestionará los procesos simultáneos (*Threads)* para ejecutar diferentes componentes electrónicos a la vez. Para ello, la clase "Padre" Hilo tiene cuatro clases "Hijos" en los que harán posible la ejecución de cada módulo por separado: PIR, Buzzer, IR_Remote y Server.

La **clase Hilo_Pir** es quien ejecuta en paralelo el módulo PIR. Para ello, hace uso de la **clase PIR** que es la que se encarga de gestionar de la detección de alguna presencia.

La **clase Hilo_Buzzer** es quien ejecuta en paralelo el módulo Buzzer. La **clase Buzzer** permite gestionara el zumbido que ha de emitir.

La **clase Hilo_IR_Remote** realiza el subproceso del módulo IR_Remote. La **clase IR_Remote** es la que se encargará de interpretar las ondas infrarrojas recibidas por el modulo IR Remote para activar o desactivar el sistema.

La **clase Hilo_Server** gestiona el proceso de ejecutar el servidor web para realizar el video streaming transmitida por la Pi Camera. Para ello es necesaria la **clase Server** que hará uso del framework Flask para la creación de la aplicación web y así poder acceder y visualizar en tiempo real lo que perciba la Pi Camera.

Por último, la **clase Servo_Motor** es la que se encarga de ejercer una fuerza sobre el servo motor para poder rotar la Pi Camera.

A4. PUBLIC KEY

En este apéndice se explica la autentificación public key entre las dos Raspberry'S Pi. En esta apéndice, la Raspberry Pi que ejerce de sistema de seguridad se llamará *Raspberry Pi #A* y la Raspbery Pi que ejerce de servidor web, *Raspberry Pi #B.*

Para la transmisión de las imágenes y vídeos que envía la Raspberry Pi #A, que ejerce de sistema de videovigilacia a la Raspberry Pi #B, que ejerce de servidor web, se hace a través del protocolo *SCP*. El protocolo SCP permite la transferencia segura de archivos entre dos host. Para que esta transferencia sea segura y autenticada se hace uso del protocolo *SSH.* El protocolo SSH sirve para acceder a maquinas remotas a través de la red, en este caso, a la Raspberry Pi #B que actúa de servidor web. Todo lo que se transmita por SSH va encriptado.

El protocolo SSH usa autentificación, esto quiere decir, que la Raspberry Pi #B le pedirá en cada conexión que solicite la Raspberry Pi #A para el envío de la información una autentificación. Para evitar que la Raspberry Pi #A se conecte la Raspberry Pi #B con autentificación se hará uso de una clave, una clave *RSA*.

Para ello, primeramente se ha de generar un conjunto de claves de cifrado asimétrico empleando el algoritmo RSA en la Raspberry Pi #A. En este momento, hay generados dos claves: clave privada y clave pública. La clave privada solo será conocida por la Raspberry Pi #A, en cambio, la clave pública la ha de compartir con la Raspberry Pi #B.

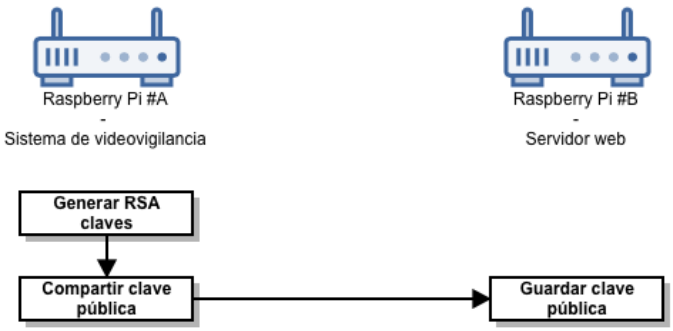

Figura 9. Diagrama de Gantt

Una vez que se ha compartido la clave pública con la Raspberry Pi #B, el proceso para que la Raspberry Pi #A acceda a la Raspberry Pi #B mediante SSH es la siguiente:

La Raspberry Pi #A hace uso de RSA para acceder a la Raspberry Pi #B. Para ello, envía su clave pública. Cuando la Raspberry Pi #B recibe dicha clave pública realiza una búsqueda en el fichero que contiene todas las claves públicas. Si encuentra la clave pública recibida cifrará un número aleatorio a partir de esa clave pública y se la enviará a la Raspberry Pi #A. En el caso de que no encuentre la clave pública en su fichero de claves denegara el acceso. Una vez enviado el número aleatorio cifrado, la Raspberry Pi #A recibirá el mensaje (el cual contiene el número cifrado) y a partir de su clave pública dará paso al descifrado. Una vez haya descifrado el mensaje (dando lugar a un número) lo enviará a la Raspberry Pi #B. La Raspberry Pi #B comprueba que ese número devuelto por parte de la Raspberry Pi #A coincide con el número aleatorio que mandó cifrado. En el caso de que los dos números coincidan, permitirá el acceso a la Raspberry Pi #A y así podrá hacer la transferencia de la imagen y video que haya captado el sistema. En caso de no coincidir, se deniega el acceso. En la Figura 10 se muestra todo el proceso.

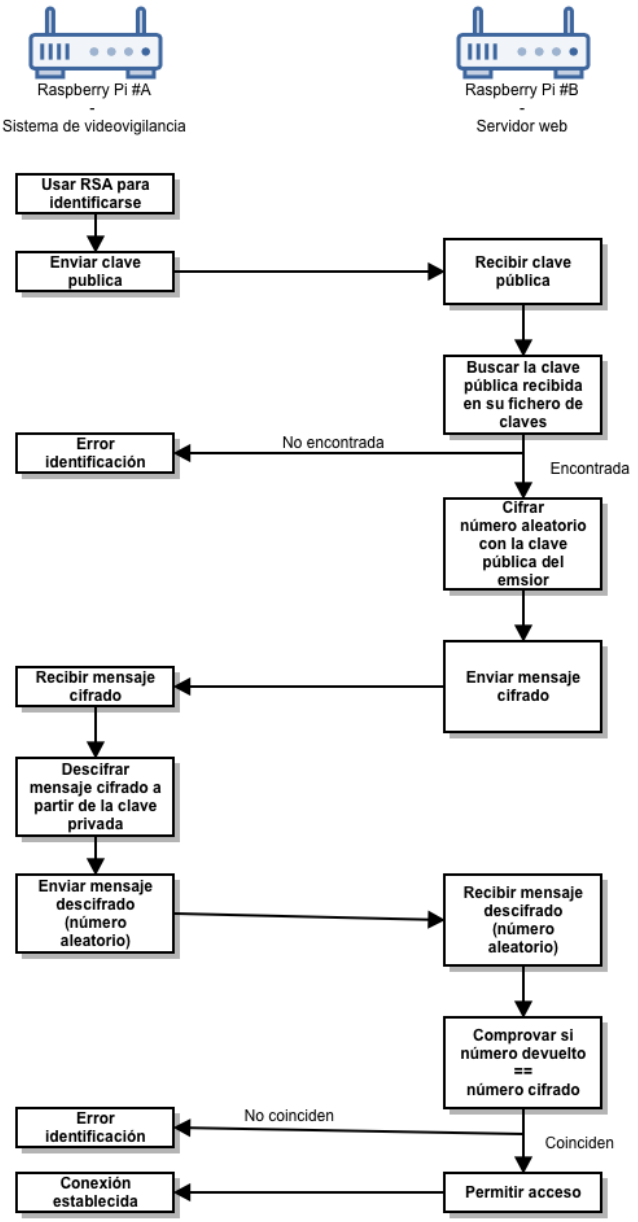

Figura 10. Diagrama de Gantt

A5. ARQUITECTURA SISTEMA DE VIDEOVIGILANCIA Y SERVIDOR WEB CON SOCKETS

En este apéndice se detalla la arquitectura de comunicación entre el sistema de videovigilancia y el servidor web para hacer posible la implementación del *video streaming* por parte del sistema de videovigilancia en el servidor web.

Por un lado tenemos el sistema de videovigilancia que será el encargado de realizar el *video streaming*. Por otro, el servidor web que contiene la plataforma web en la que deberá visualizar el video streaming. Al tratarse de dos programas completamente diferentes surge la necesidad de hacer uso de los sockets para realizar una conexión entre ellas dos.

Un socket es un túnel de comunicación que ayuda a que dos programas se comuniquen. En este caso, se trata de una arquitectura cliente-servidor. El sistema de videovigilancia (cliente) enviara el "video streaming al servidor web (servidor) para que se pueda visualizar en la plataforma web.

Para su implementación, cada parte tiene una serie de características. El sistema de videovigilancia (cliente) ha de definir una conexión utilizando el socket creado a partir de una dirección IP y puerto hacia donde se iniciara la conexión, es decir, al servidor web. Después, debe solicitar el inicio de una conexión con el servidor el cual le responderá diciéndole que está conectado correctamente. Una vez establecida la conexión, es posible el intercambio de comunicación. Respecto al servidor web (servidor), se crea una conexión utilizando el socket, indicando una dirección IP y puerto por el que se iniciará la conexión del socket. A partir de aquí, se iniciaría un bucle infinito en el que estará todo el tiempo en modo escucha del cliente.

Una vez que tenemos las dos partes configuradas a través de los sockets, es posible él envió de información. Para ello, en la parte del sistema de videovigilancia se ha de transformar los fotogramas capturados por el sistema de videovigilancia en binario. Esto es necesario ya que la información se envia en forma binaria por lo que se ha de encargar el sistema de videovigilancia en transformar la información a binario. Una vez hecho eso, se envía el paquete y el servidor web es quien lo recibe. Una vez el servidor web haya recibió el paquete, se dispone abrirlo y a transformar de nuevo a su tipo original.

Esto hace posible que mientras el sistema de videovigilancia esté realizando un video streaming, por la parte del servidor web se podrá visualizar en la plataforma web dicho video streaming. En la Figura 11 se detalla la comunicación para el envió de los datos transmititos por la Pi Camera de una Raspberry a otra.

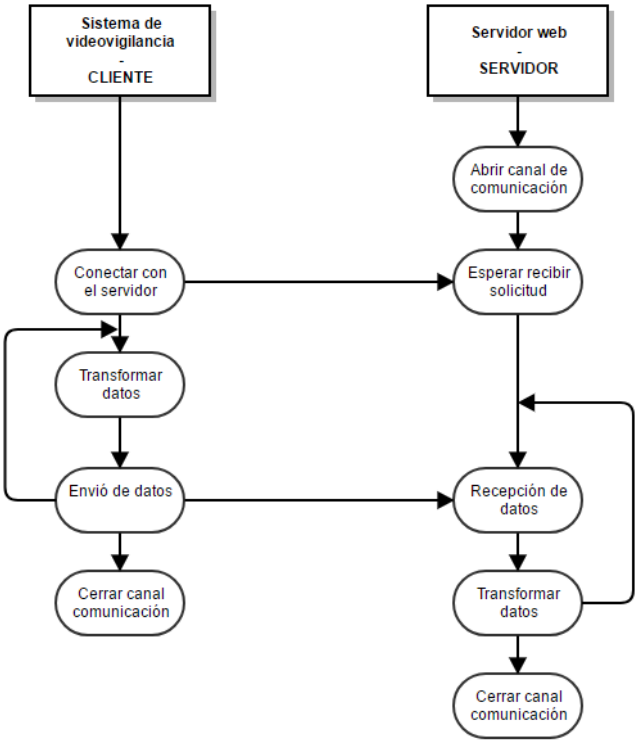

Figura 11. Diagrama de Gantt